

Customizing and Deploying Open-source LLMs



Laurentiu Raducu

Founder of bitheap

@bitheap_tech | www.bitheap.ch





The AI community building the future.

The platform where the machine learning community collaborates on models, datasets, and applications.

[Explore AI Apps](#)[or Browse 1M+ models](#)

Tasks Libraries Datasets Languages Licenses Other

Filter Tasks by name

Multimodal

Text-to-Image Image-to-Text
Text-to-Video Visual Question Answering
Document Question Answering Graph Machine Learning

Computer Vision

Depth Estimation Image Classification
Object Detection Image Segmentation
Image-to-Image Unconditional Image Generation
Video Classification Zero-Shot Image Classification

Natural Language Processing

Text Classification Token Classification
Table Question Answering Question Answering
Zero-Shot Classification Translation
Summarization Conversational
Text Generation Text2Text Generation
Sentence Similarity

Audio

Text-to-Speech Automatic Speech Recognition
Audio-to-Audio Audio Classification
Voice Activity Detection

Tabular

Tabular Classification Tabular Regression

Reinforcement Learning

Reinforcement Learning Robotics

Models 469,541 Filter by name

meta-llama/Llama-2-70b
Text Generation • Updated 4 days ago • ↓ 25.2k • ❤ 64

stabilityai/stable-diffusion-xl-base-0.9
Updated 6 days ago • ↓ 2.01k • ❤ 393

openchat/openchat
Text Generation • Updated 2 days ago • ↓ 1.3k • ❤ 136

illyasviel/ControlNet-v1-1
Updated Apr 26 • ✓ 1.87k

cerspense/zeroscope_v2_XL
Updated 3 days ago • ↓ 2.66k • ❤ 334

meta-llama/Llama-2-13b
Text Generation • Updated 4 days ago • ↓ 328 • ❤ 64

tiiuae/falcon-40b-instruct
Text Generation • Updated 27 days ago • ↓ 288k • ❤ 899

WizardLM/WizardCoder-15B-V1.0
Text Generation • Updated 3 days ago • ↓ 12.5k • ❤ 332

CompVis/stable-diffusion-v1-4
Text-to-Image • Updated about 17 hours ago • ↓ 448k • ❤ 5.72k

stabilityai/stable-diffusion-2-1
Text-to-Image • Updated about 17 hours ago • ↓ 782k • ❤ 2.81k

Salesforce/xgen-7b-8k-inst
Text Generation • Updated 4 days ago • ↓ 6.18k • ❤ 57



Hugging Face

Models Datasets Spaces Community Docs Enterprise Pricing

g google/gemma-3-270m like 793 Follow Google 29k

Text Generation Transformers Safetensors gemma3_text gemma3 gemma google text-generation-inference arxiv:35 papers License: gemma

Model card Files and versions Community 19 Train Deploy Use this model

main gemma-3-270m Go to file 2 contributors History: 7 commits Contribute

osanseviero	Update README.md	9b0cfec	VERIFIED	about 1 month ago	
.gitattributes	Safe	1.57 kB	uploading the PT weights for Gemma 3 270m	about 1 month ago	
README.md	Safe	28.3 kB	Update README.md	about 1 month ago	
added_tokens.json	Safe	35 Bytes	uploading the PT weights for Gemma 3 270m	about 1 month ago	
config.json	Safe	1.35 kB	uploading the PT weights for Gemma 3 270m	about 1 month ago	
generation_config.json	Safe	133 Bytes	Removing chat template	about 1 month ago	
model.safetensors	Safe ↗	536 MB	LFS	uploading the PT weights for Gemma 3 270m	about 1 month ago
special_tokens_map.json	Safe	662 Bytes	uploading the PT weights for Gemma 3 270m	about 1 month ago	
tokenizer.json	Safe	33.4 MB	LFS	uploading the PT weights for Gemma 3 270m	about 1 month ago
tokenizer.model	Safe	4.69 MB	LFS	uploading the PT weights for Gemma 3 270m	about 1 month ago
tokenizer_config.json	Safe	1.16 MB	uploading the PT weights for Gemma 3 270m	about 1 month ago	

```
{ "_sliding_window_pattern": 6, "architectures": [ "Gemma3ForCausalLM" ], "attention_bias": false, "attention_dropout": 0.0, "attn_logit_softcapping": null,
```

Config.json

```
{
  "hidden_size": 4096,
  "attention_bias": false,
  "num_hidden_layers": 32,
  "attn_logit_softcapping": null,
  ...
  "transformers_version": "4.55.0.dev0",
  "use_bidirectional_attention": false,
  "use_cache": true,
  "vocab_size": 262144
}
```



Hugging Face

Models Datasets Spaces Community Docs Enterprise Pricing

g google/gemma-3-270m like 793 Follow Google 29k

Text Generation Transformers Safetensors gemma3_text gemma3 gemma google text-generation-inference arxiv:35 papers License: gemma

Model card Files and versions Community 19 Train Deploy Use this model

main gemma-3-270m Go to file 2 contributors History: 7 commits Contribute

osanseviero	Update README.md	9b0cfec	VERIFIED	about 1 month ago	
.gitattributes	Safe	1.57 kB	uploading the PT weights for Gemma 3 270m	about 1 month ago	
README.md	Safe	28.3 kB	Update README.md	about 1 month ago	
added_tokens.json	Safe	35 Bytes	uploading the PT weights for Gemma 3 270m	about 1 month ago	
config.json	Safe	1.35 kB	uploading the PT weights for Gemma 3 270m	about 1 month ago	
generation_config.json	Safe	133 Bytes	Removing chat template	about 1 month ago	
model.safetensors	Safe ↗	536 MB	LFS	uploading the PT weights for Gemma 3 270m	about 1 month ago
special_tokens_map.json	Safe	662 Bytes	uploading the PT weights for Gemma 3 270m	about 1 month ago	
tokenizer.json	Safe	33.4 MB	LFS	uploading the PT weights for Gemma 3 270m	about 1 month ago
tokenizer.model	Safe	4.69 MB	LFS	uploading the PT weights for Gemma 3 270m	about 1 month ago
tokenizer_config.json	Safe	1.16 MB	uploading the PT weights for Gemma 3 270m	about 1 month ago	

Model Modifications



Config parameters to edit

- max_length
- temperature
- top_p



Config.json

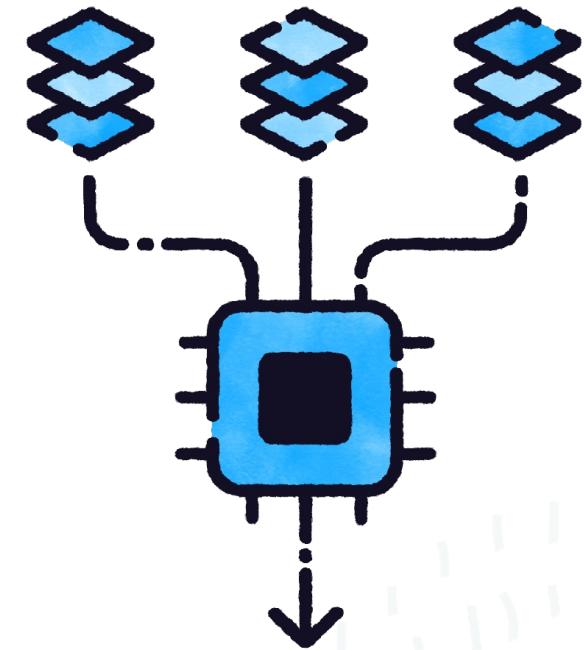
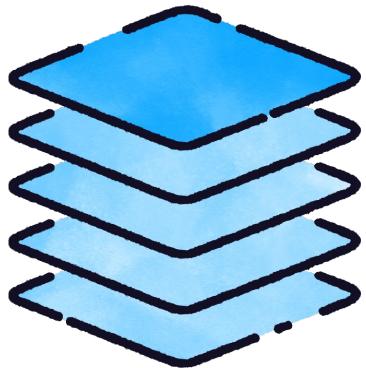
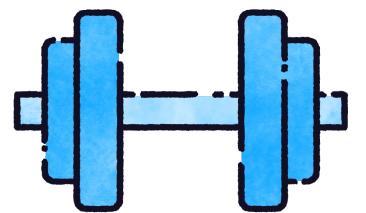
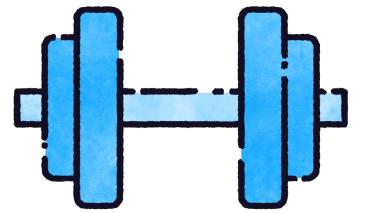
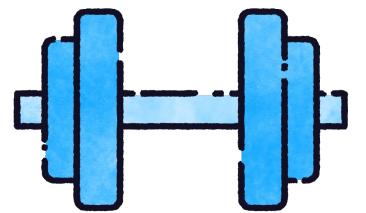
```
{  
    "hidden_size": 4096,  
    "attention_bias": false,  
    "num_hidden_layers": 32,  
    "attn_logit_softcapping": null,  
    ...  
    "transformers_version": "4.55.0.dev0",  
    "use_bidirectional_attention": false,  
    "use_cache": true,  
    "vocab_size": 262144  
}
```



Config.json

```
{  
    "hidden_size": 4096,  
    "attention_bias": false,  
    "num_hidden_layers": 24,  
    "attn_logit_softcapping": null,  
    ...  
    "transformers_version": "4.55.0.dev0",  
    "use_bidirectional_attention": false,  
    "use_cache": true,  
    "vocab_size": 262144  
}
```





Key Points



Open-source models have

- weights
- configs
- tokenizers



Understanding LLM Architecture and Customization Options



LLM Architecture

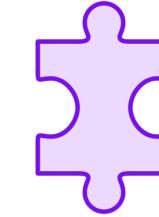
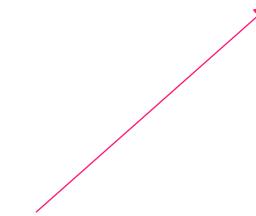
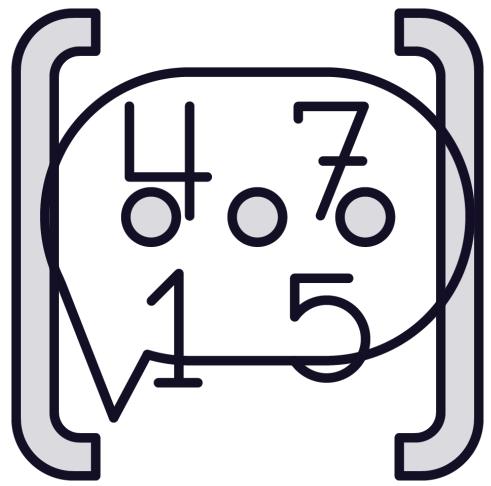
Tokenizer



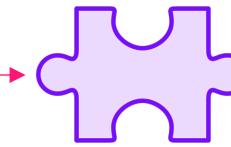
tokenizer.json

```
{  
  "version": "1.0",  
  "truncation": null,  
  "padding": null,  
  "added_tokens": [  
    {  
      "id": 0,  
      "content": "<pad>",  
      "single_word": false,  
      "lstrip": false,  
      ...  
    }  
  ]  
}
```

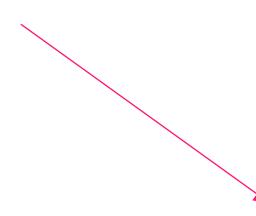




Token



Token



Token



**Choosing a different
tokenizer can break a
model.**



LLM Architecture

Tokenizer

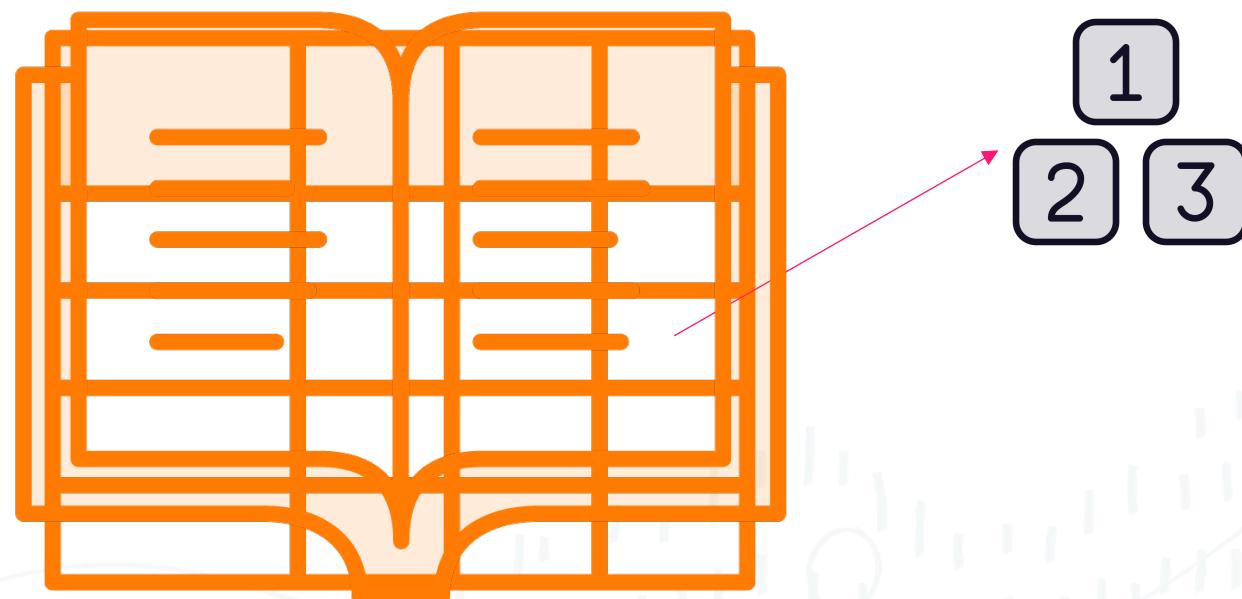
Embedding Layers



wte.weight

$$[\circ \circ \circ]$$

wpe.weight



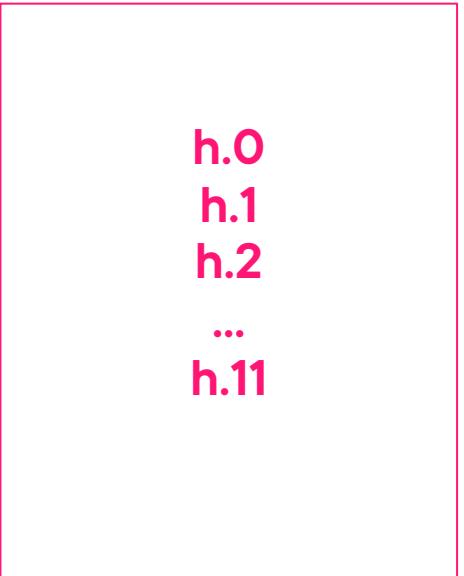
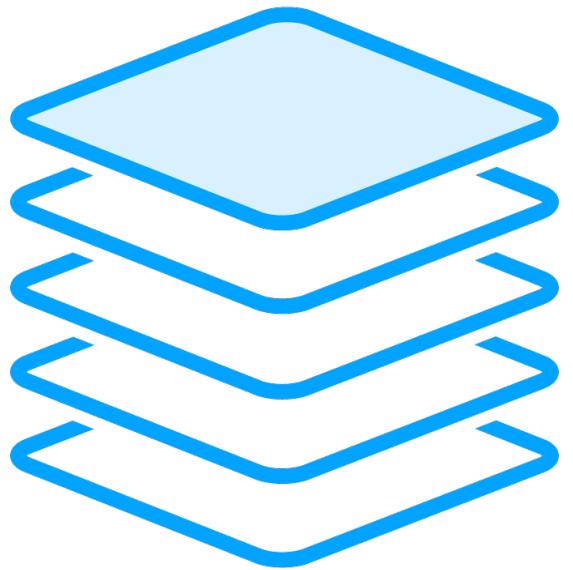
LLM Architecture

Tokenizer

Embedding Layers

Transformer Blocks





Transformer Block Components

Multi-head attention
mechanism

Feed-forward
network

Layer normalization
and residual
connections



**The chef ate the pizza
because *he* was hungry.**



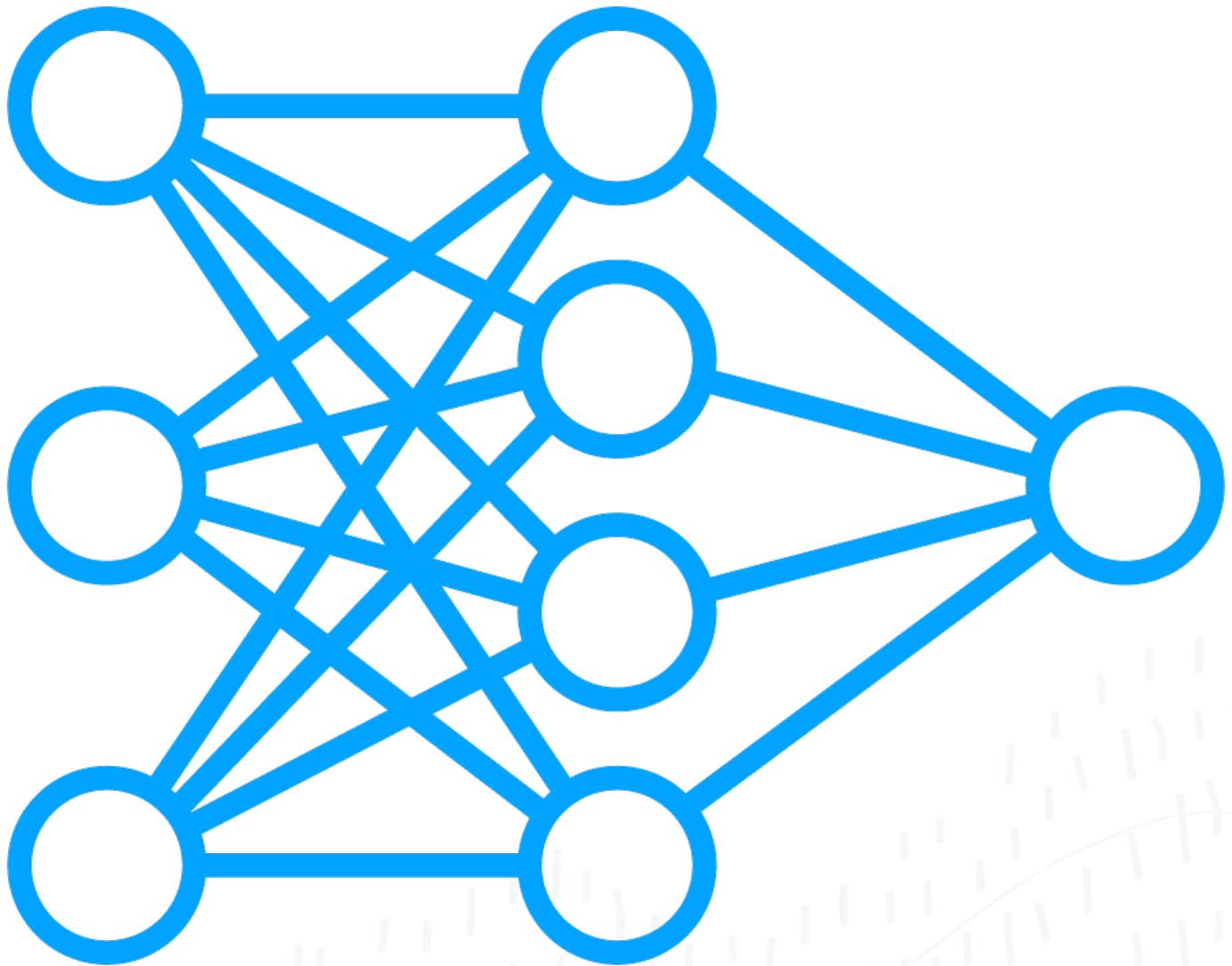
Transformer Block Components

Multi-head attention
mechanism

Feed-forward
network

Layer normalization
and residual
connections





LLM Architecture

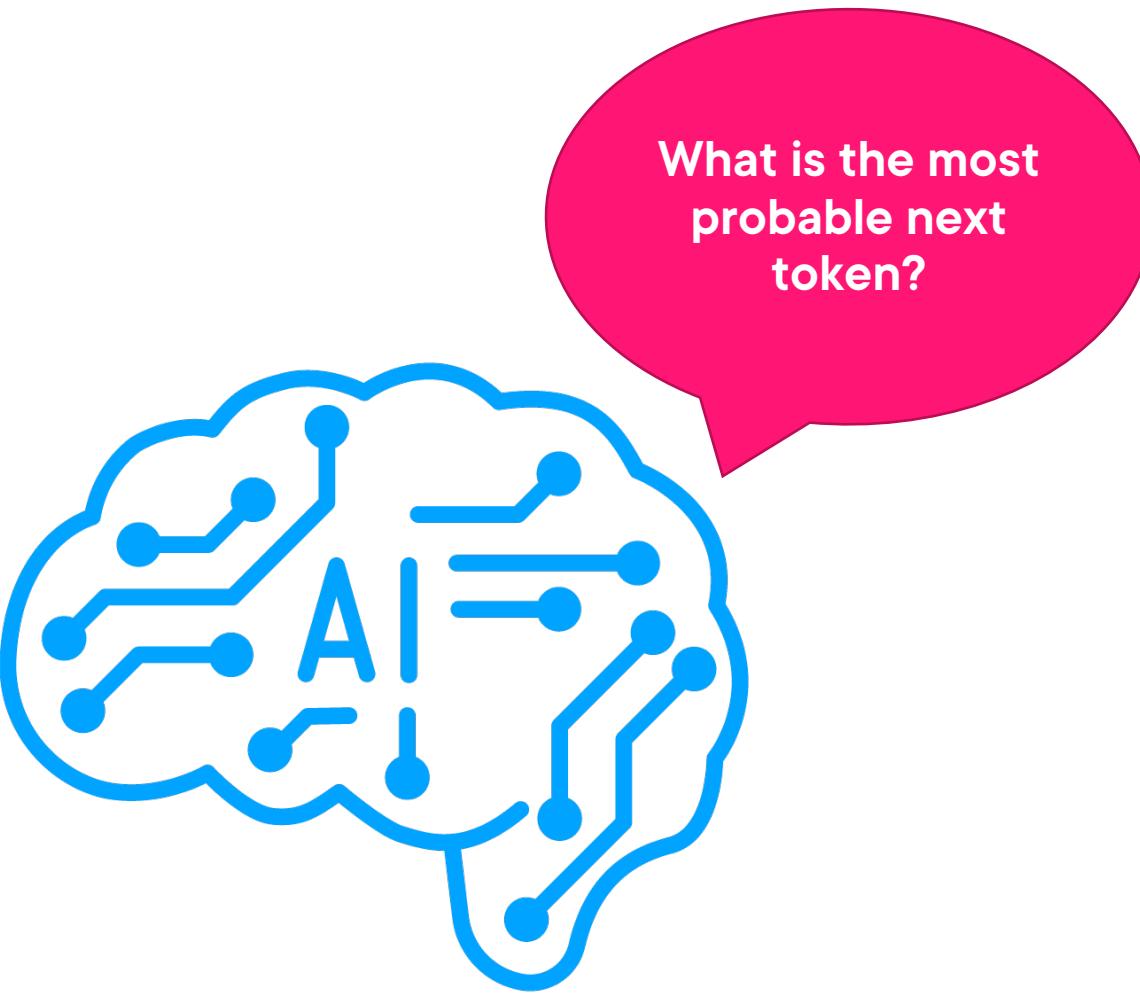
Tokenizer

Embedding Layers

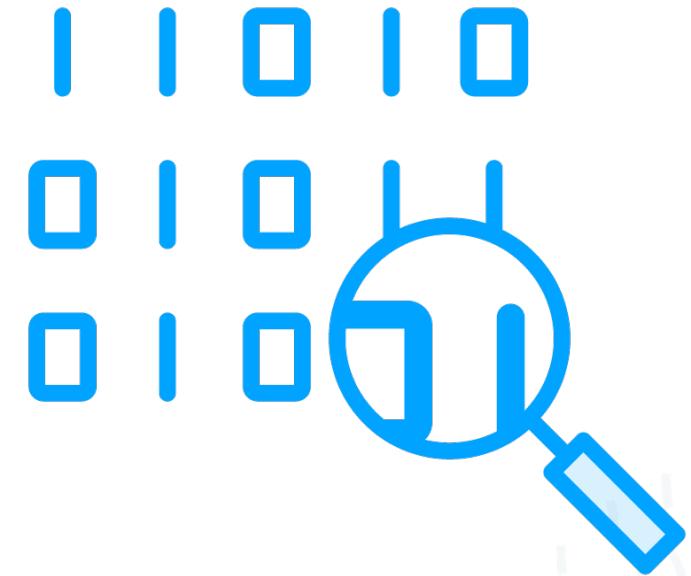
Transformer Blocks

Output Layer





What is the most
probable next
token?





Prompt engineering

- Requires no change to the model
- Done by customizing the input prompt
- Powerful and free, but limited

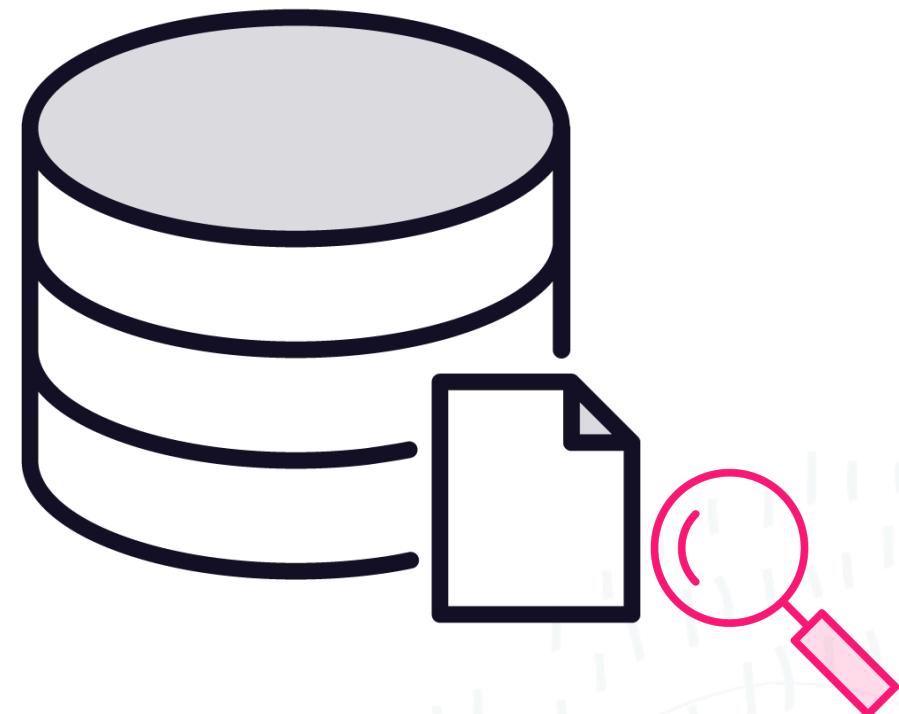
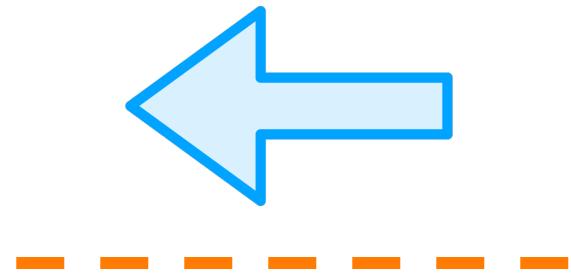
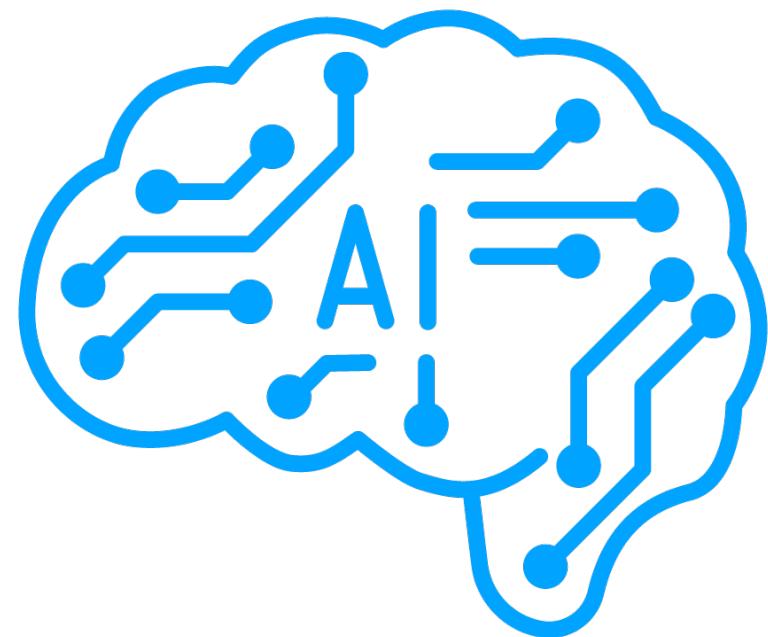
Modifying inference parameters

- Change parameters like temperature, top_p, top_k

Retrieval-augmented generation

- Architectural setup







Prompt engineering

- Requires no change to the model
- Done by customizing the input prompt
- Powerful and free, but limited

Modifying inference parameters

- Change parameters like temperature, top_p, top_k

Retrieval-augmented generation

- Architectural setup

Architectural changes

- Modifying config.json file and weights



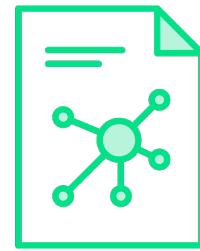
Fine-tuning Open-source LLMs



Summary

Fine-tuning main approaches
How to select and prepare a dataset



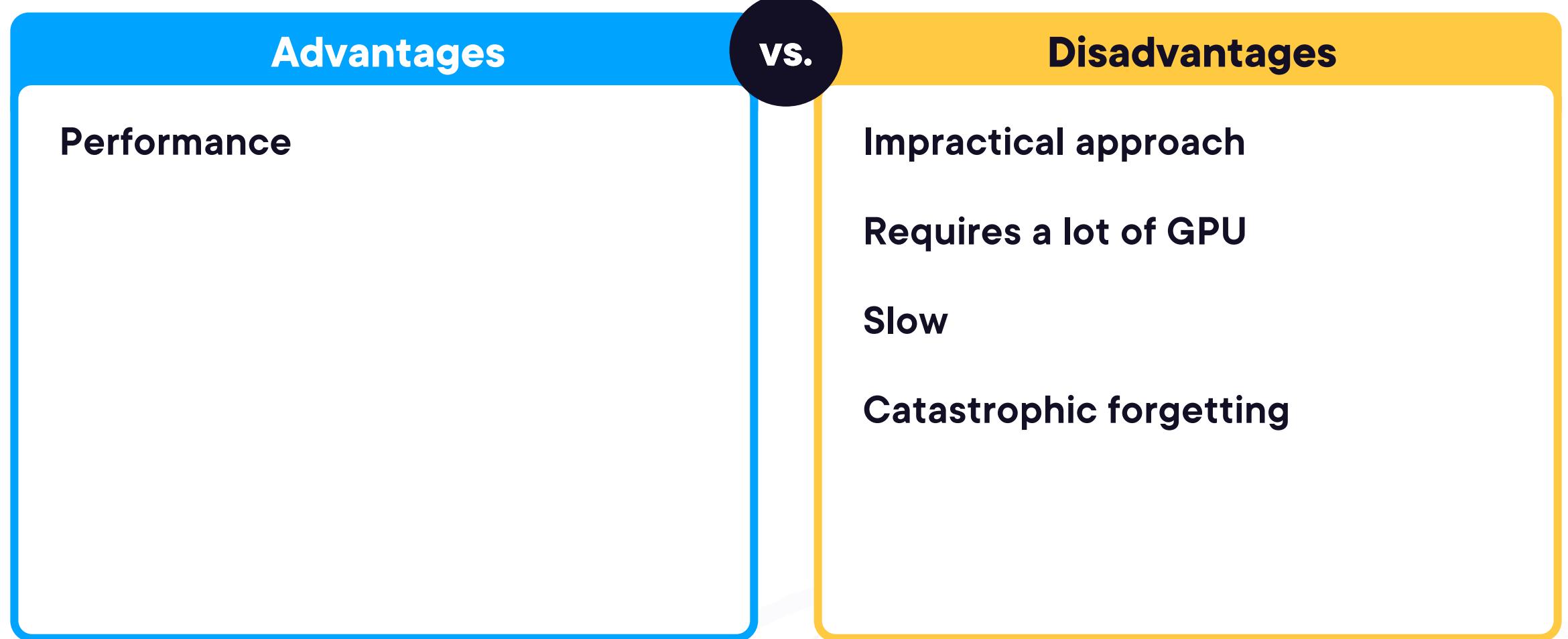


Fine-tuning Types

Full fine-tuning



Full Fine-tuning Features

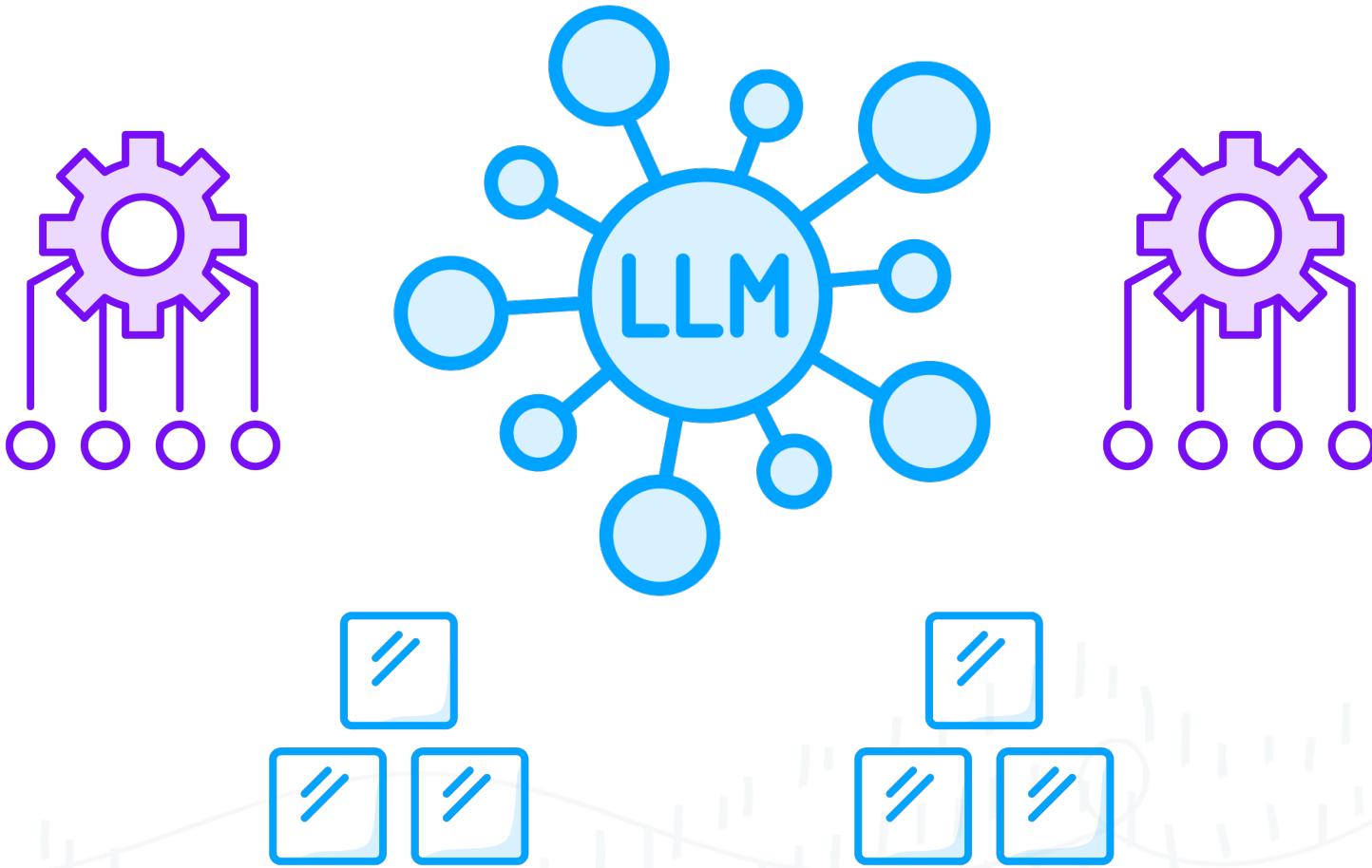


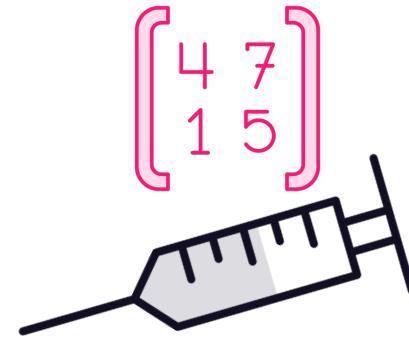
Fine-tuning Types

Full fine-tuning

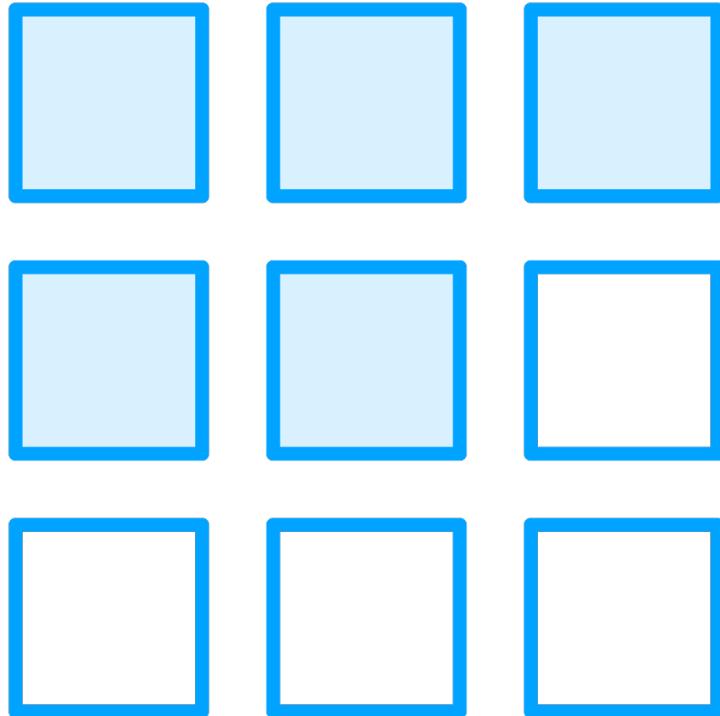
Parameter-efficient fine-tuning







LoRA Benefits



- Drastic reduction in cost
- Speed
- Portability
- No forgetting



**Selecting and preparing
your dataset is arguably the
most important part of the
entire process.**



Data Quality Elements

Format



Customer Service Bot Data

```
{  
    "instruction": "A customer writes: 'My login doesn't work anymore.' How would  
you respond?",  
    "output": "I'm sorry to hear you're having trouble. Let's get that fixed. To  
help you, could you please tell me your username or the email associated with  
your account?"  
}
```

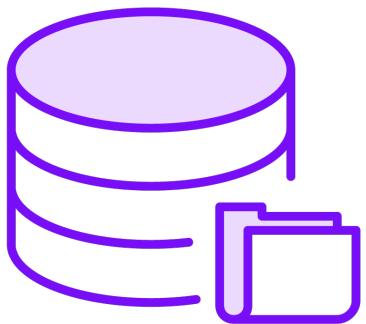


Data Quality Elements

Format

Size





Data Quality Elements

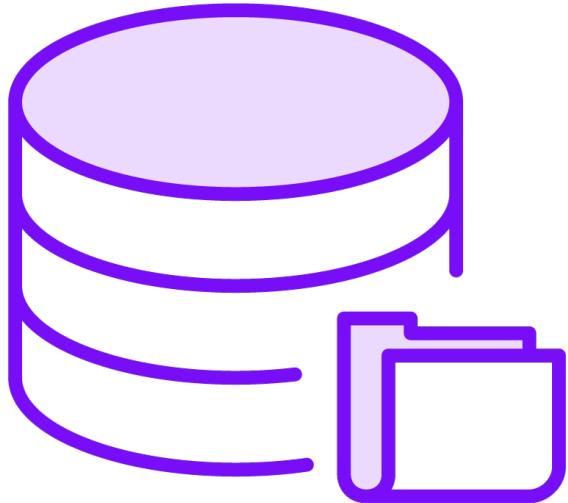
Format

Size

Quality



Data Quality

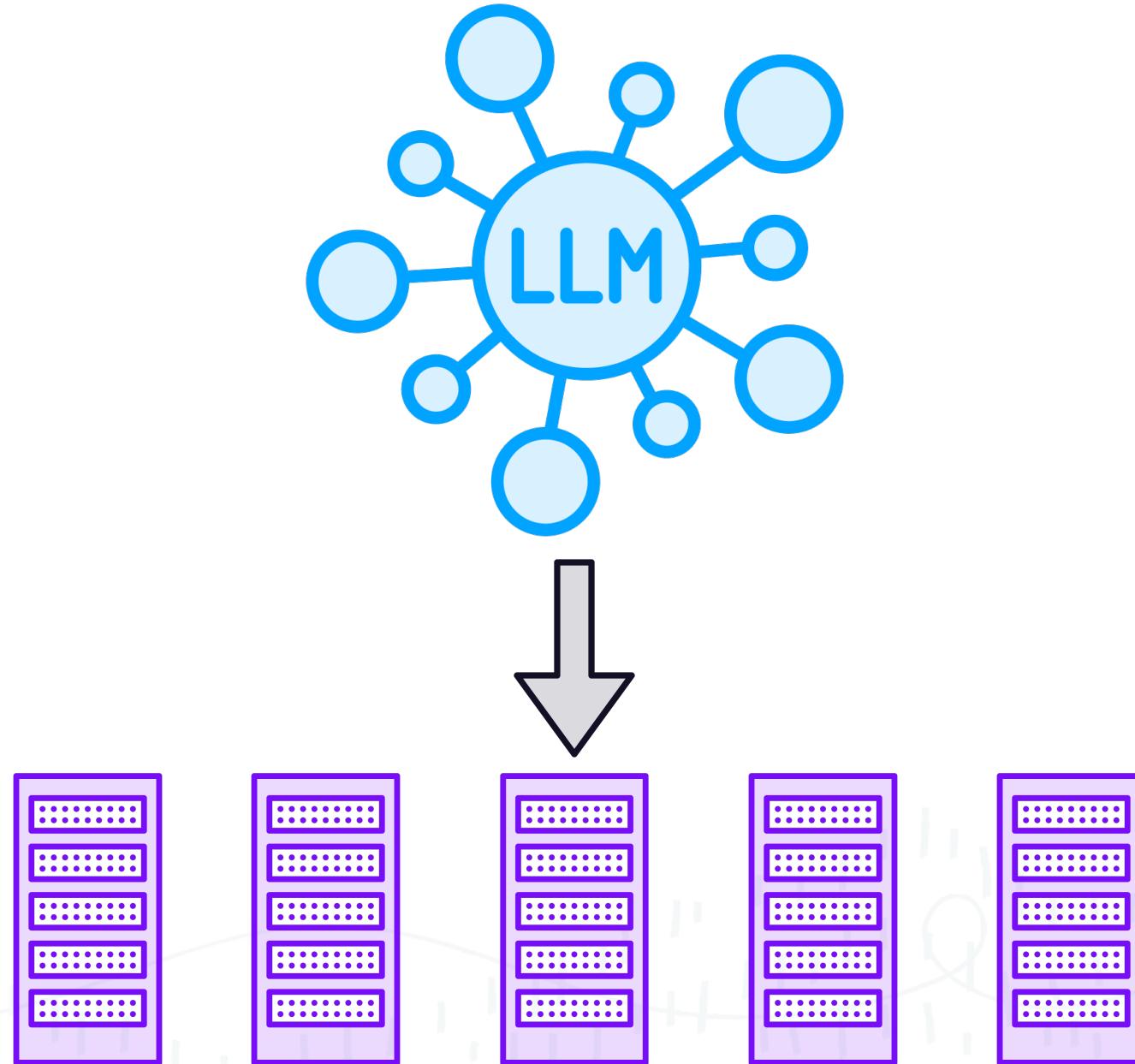


- Accuracy
- Consistency
- Representation

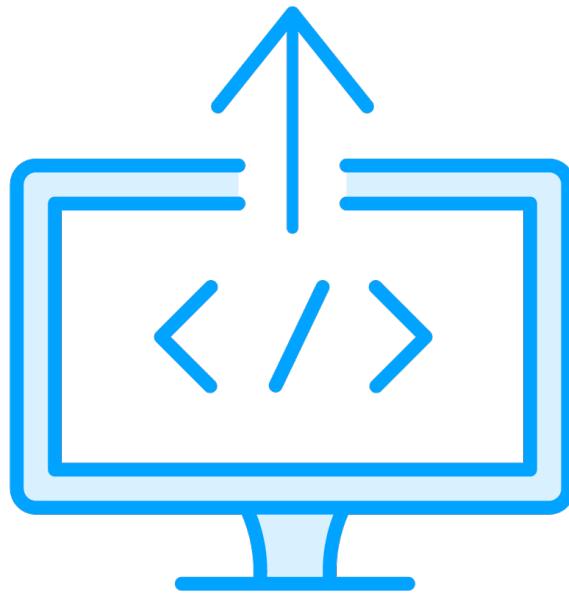


Deploying LLMs on Your Own Infrastructure





Local Deployment Benefits



- Full data privacy
- Reliability
- Predictable costs





OpenAI and Ollama partner to launch [gpt-oss](#)

**Chat & build with
open models**

Download

Available for macOS,
Windows, and Linux

```
curl -fsSL  
https://ollama.com/install.sh | sh
```

◀ **Install Ollama on Linux**

```
ollama run llama3
```

◀ **Run the Llama3 model**

```
curl  
http://localhost:11434/api/generate -  
d '{  
  "model": "llama3",  
  "prompt": "Write me a short,  
beautiful poem about the sunrise."  
}'
```

◀ **Instruct Llama3 to write a poem**



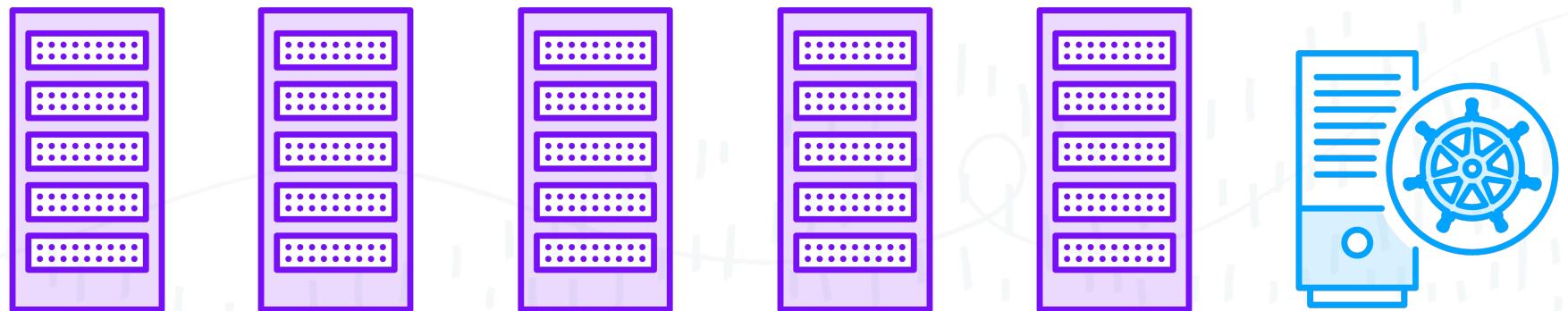
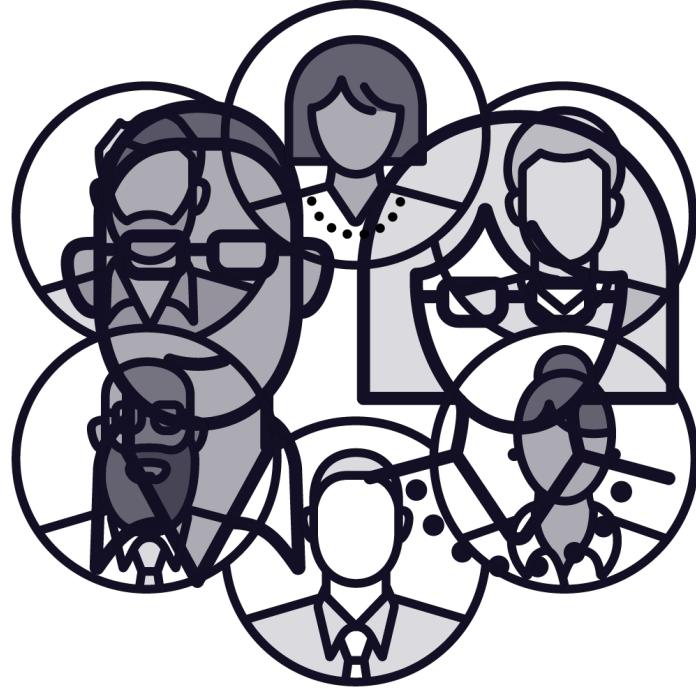
Local Deployment Hardware Requirements



For a 7-billion-parameter model:

- GPU with at least 14 GB of VRAM
- CPU works, but much slower





Industry-specific Customization Demonstration



THE FUTURE WAS FIVE MINUTES AGO.

We were there and it is going to be epic

FUTURE CONFERENCE

SIGN UP TODAY

Tickets are selling fast
get yours today and be
there in the future

DARK ENERGY

BRIGHTENS THE WORLD



ROBOTICS

THEY ARE HERE AND THEY PLAY

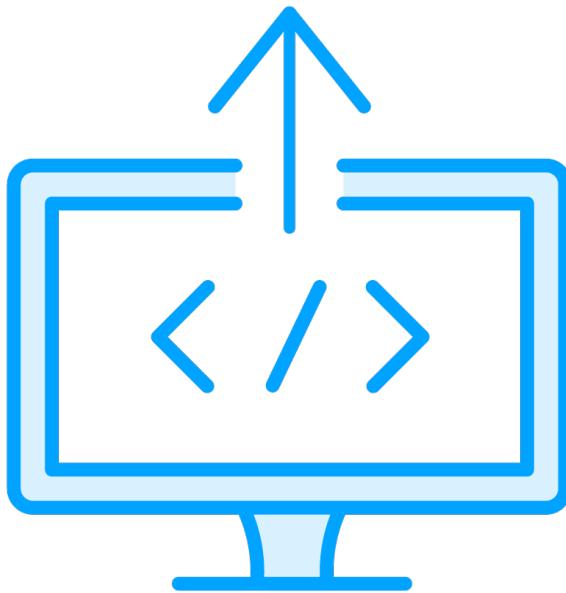


STRANGERS RISE

ASCENSION GOES TECHNICAL



Fine-tuning Process



- Defining the personality
- Preparing the data
- Fine-tuning the model
- Building an interface

Defining the Persona and Data Preparation

Patient, professional with a friendly edge

Can explain Globomantics' subscriptions

Able to troubleshoot

It must always defer to human agents for complex issues



Training File

```
[  
 {  
   "instruction": "A customer writes: 'I think I want to upgrade my plan. What's the difference between Pro and Enterprise?'",  
   "output": "Hello! I'd be happy to explain the differences. Our Pro plan is designed for individual professionals and includes priority support. The Enterprise plan adds features for teams, like centralized billing and admin controls, and dedicated account management. May I ask about your team size to see if Enterprise is a good fit?"  
 },  
 {  
   "instruction": "A frustrated customer says: 'I can't log in to the application! This is ridiculous.'",  
   "output": "I'm really sorry you're having trouble logging in. I know how frustrating that can be. Let's get it sorted out. First, could you confirm if you're seeing any specific error message? In the meantime, the quickest fix is often to reset your password using the 'Forgot Password' link on the login page."  
 },  
 {  
   "instruction": "A customer asks: 'When will my favorite feature be added?'",  
   "output": "That's a great question! I don't have access to our internal product roadmap, so I can't give you a specific timeline for new features. What I can do is pass your feedback about wanting this feature directly to our product team. They review all customer requests regularly."  
 }  
 ]
```

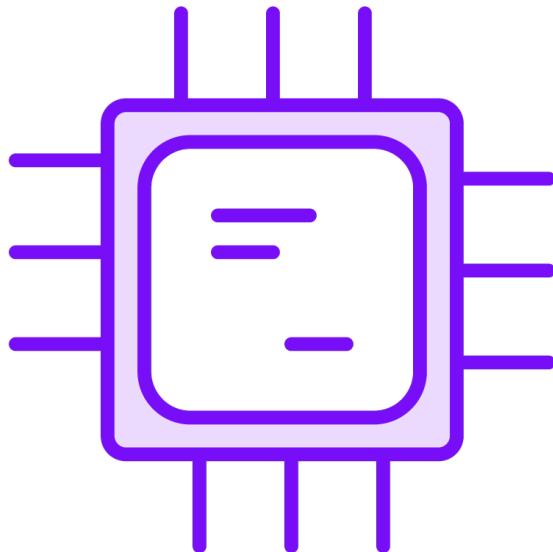


Training File

```
[  
...  
{  
  "instruction": "A frustrated customer says: 'I can't log in to the  
application! This is ridiculous.'",  
  "output": "I'm really sorry you're having trouble logging in. I know how  
frustrating that can be. Let's get it sorted out. First, could you confirm if  
you're seeing any specific error message? In the meantime, the quickest fix is  
often to reset your password using the 'Forgot Password' link on the login  
page."  
},  
...  
]
```



Fine-tuning Prerequisites



- GPU – 16GB VRAM for an 8b model
- Cloud GPU – Lambda Labs, RunPod or AWS
- Python



```
python -m venv unsloth-env  
source unsloth-env/bin/activate # On Windows: unsloth-env\Scripts\activate  
  
pip install "unsloth[colab-new] @ git+https://github.com/unslotha/unsloth.git"  
  
pip install --no-deps transformers datasets trl accelerate huggingface_hub
```

Preparing the Environment

Commands for setting up a clean Python environment for model fine-tuning.



```
from unsloth import FastLanguageModel
import torch
from datasets import load_dataset
from trl import SFTTrainer
from transformers import TrainingArguments

model, tokenizer =
FastLanguageModel.from_pretrained(
    model_name = "unsloth/Meta-Llama-3.1-
8B",
    max_seq_length = 2048,
    dtype = None,
    load_in_4bit = True,
)
```

◀ Dependencies imports

◀ Loading the base model with Unsloth optimization



```
model = FastLanguageModel.get_peft_model(  
    model,  
    r = 16,  
    target_modules = ["q_proj", "k_proj",  
"v_proj", "o_proj", "gate_proj", "up_proj",  
"down_proj",],  
    lora_alpha = 16,  
    lora_dropout = 0,  
    bias = "none",  
    use_gradient_checkpointing = True,  
    random_state = 3407,  
)
```

```
dataset = load_dataset("json",  
data_files="globomantics_training_data.json",  
split="train")
```

◀ Adding LoRA adapters to the model.

◀ Loading the data set.



```
def formatting_prompts_func(examples):
    instructions = examples["instruction"]
    outputs = examples["output"]
    texts = []
    for instruction, output in zip(instructions,
outputs):

        text =
f"<|start_header_id|>user<|end_header_id|>\n\n{in
struction}<|eot_id|><|start_header_id|>assistant<
|end_header_id|>\n\n{output}<|eot_id|>"
        texts.append(text)
    return { "text" : texts, }
dataset = dataset.map(formatting_prompts_func,
batched = True)
```

◀ Formatting the training data



```
training_args = TrainingArguments(  
    output_dir = "outputs",  
    per_device_train_batch_size = 2,  
    gradient_accumulation_steps = 4,  
    warmup_steps = 5,  
    num_train_epochs = 3,  
    learning_rate = 2e-4,  
    fp16 = not torch.cuda.is_bf16_supported(),  
    bf16 = torch.cuda.is_bf16_supported(),  
    logging_steps = 1,  
    optim = "adamw_8bit",  
    weight_decay = 0.01,  
    lr_scheduler_type = "linear",  
    seed = 3407,  
    save_strategy = "epoch",  
)  
  
trainer = SFTTrainer(  
    model = model,  
    tokenizer = tokenizer,  
    train_dataset = dataset,  
    dataset_text_field = "text",  
    max_seq_length = 2048,  
    args = training_args,  
)  
  
trainer.train()  
  
model.save_pretrained("globo_assist_lora_adapter")
```

◀ Configuring the trainer's arguments

◀ Initializing the trainer

- ◀ Training the model
- ◀ Saving the adapter as
globo_assist_lora_adapter



Modelfile Definition

Defining the file to load into Ollama

Modelfile

```
FROM /path/to/llama3.1-8b  
  
ADAPTER /path/to/globo-assist-lora-adapter.safetensors  
  
TEMPLATE "{{ .System }} {{ .Prompt }}"  
  
SYSTEM "You are GloboAssist, a helpful and professional customer support chatbot for Globomantics Software Inc. You help users with account and subscription questions."  
  
PARAMETER temperature 0.7
```

```
> ollama create globo-assist -f Modelfile
```



```
import streamlit as st
import requests

st.set_page_config(page_title="GloboAssist",
page_icon="🤖")
st.title("GloboAssist Support Bot")

OLLAMA_URL =
"http://localhost:11434/api/generate"

if "messages" not in st.session_state:
    st.session_state.messages = [ ]

for message in st.session_state.messages:
    with st.chat_message(message["role"]):
        st.markdown(message["content"])
```

◀ Dependencies imports

◀ Sets the page title and icon

◀ Defines the URL where Ollama runs

◀ Initializes the chat history

◀ Displays the user message in the chat history



```
if prompt := st.chat_input("How can I help you today?"):
    st.chat_message("user").markdown(prompt)
    st.session_state.messages.append({"role": "user",
"content": prompt})

request_data = {
    "model": "globo-assist",
    "prompt": prompt,
    "stream": False
}

try:
    response = requests.post(OLLAMA_URL,
json=request_data)
    response.raise_for_status()
    full_response = response.json()
    assistant_msg = full_response['response']

except Exception as e:
    assistant_msg = f"Sorry, I'm experiencing a
technical issue: {e}"

with st.chat_message("assistant"):
    st.markdown(assistant_msg)
    st.session_state.messages.append({"role": "assistant", "content": assistant_msg})
```

◀ Reacts to the user input

◀ Adds the message to chat history

◀ Prepares the data for the Ollama API

◀ Sends the request to Ollama

◀ Displays the response and adds it to the chat history



Summary

Fine-tuned a general-purpose model

Created an interface

Deployed it on the local infrastructure

